2009

Comparison of Random Forests and Cforest: Variable Importance Measures and Prediction Accuracies

Rong Xia
Utah State University

COMPARISON OF RANDOM FORESTS AND CFOREST: VARIABLE IMPORTANCE MEASURES AND PREDICTION ACCURACIES

by

Rong Xia

A report submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Statistics

UTAH STATE UNIVERSITY
Logan, Utah

2009

Abstract

Comparison of Random Forests and Cforest: Variable Importance Measures and Prediction Accuracies

by

Rong Xia, Master of Science

Utah State University, 2009

Major Professor: Dr. Adele Cutler
Department: Mathematics and Statistics

Random forests are ensembles of trees that give accurate predictions for regression, classification and clustering problems. The CART tree, the base learner employed by random forests, has been criticized because of bias in the selection of splitting variables. The performance of random forests is suspect due to this criticism. A new implementation of random forests, Cforest, which is claimed to outperform random forests in both predictive power and variable importance measures, was developed based on Ctree, an implementation of conditional inference trees.

We address the underlying mechanism of random forests and Cforest in this report. Comparison of random forests and Cforest is presented based on simulated data. Our study shows that except for some extreme situations, with proper choice of tuning parameter values, random forests provides higher prediction accuracies and more reliable variable importance measures than Cforest.

(36 pages)

This goes to my parents, who always give me the greatest love.

Acknowledgments

I would like to thank Dr. Adele Cutler, my advisor, for her great knowledge of statistics and the time (and patience) she spent helping me with this research. I would also like to express thanks for all other helpful advice and suggestions from the professors and my colleagues at the Department of Mathematics and Statistics, Utah State University.

Rong Xia

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Although random forests is a general tool designed for regression, classification and clustering problems, for convenience, we will only focus on classification in this article.

1.1 Random Forests

Random forests were introduced by Leo Breiman in 2001 [1], and can be considered as an ensemble method that combines a large collection of trees. More explicitly, we select bootstrap samples from the original learning (training) data and fit a binary CART (Breiman et al. 1984 [2]) tree to each bootstrap sample. Random forests is obtained by voting all the trees. When fitting the trees, at each node, we randomly choose a small subset of predictor (covariate) variables and find the best split over these variables only. The splitting procedure is repeated until a certain stopping criterion (the node is pure or its size is smaller than a pre-specified value) is met. Typically the tree is grown until it is sufficiently large, with no pruning.

Let $\mathcal{L} = \{\mathbf{z}_n = (\mathbf{x}_n, y_n)\}_{n=1}^{N}$ be the learning data, $\{X_j\}_{j=1}^{p}$ be the predictor variables, Y be the response variable, and $\{\mathcal{S}_b\}_{b=1}^{B}$ be the bootstrap samples from $\mathcal{L}$. Here is an illustration of the random forests algorithm for classification [3] [4] [5] [6].

Algorithm 1 : Random Forests for Classification

---

1. For $b = 1$ to B:

   (a) Draw a bootstrap sample $\mathcal{S}_b$ from the learning data $\mathcal{L}$.

   (b) Grow a binary classification tree T_b to the bootstrapped data $\mathcal{S}_b$, by recursively repeating the following steps for each node of the tree, until the stopping criterion is met.

i. Randomly select m variables from the p predictor variables.

ii. Pick the best split from all possible splits over the m selected variables, based on the Gini Criterion.

iii. Split the node into two descendant nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point $\mathbf{x}$:

Let $\hat{C}_b(\mathbf{x})$ be the prediction of the bth tree at $\mathbf{x}$, then $\hat{C}_{rf}^B(\mathbf{x})$ = majority vote $\{\hat{C}_b(\mathbf{x})\}_1^B$.

The Gini Criterion is illustrated as follows:

Assume there are N_l observations in node l, define $D_l = \{n|$ observation $\mathbf{z}_n$ is in node $l\}$. We can estimate the proportion of class-k observations in node l as $\hat{p}_{lk} = \frac{1}{N_l}\sum_{n\in D_l} I(y_n = k)$, where $I(y_n = k)$ is the indicator function.

The *Gini Index*, an estimate of the measure of impurity in node l, is then defined as $\hat{G}_l = \sum\sum_{k\neq k'}\hat{p}_{lk}\hat{p}_{lk'}$ [7].

Suppose we split node l into two descendant nodes using a split on predictor X_j. If X_j is numerical, the split is determined by a cutpoint a_j and if X_j is categorical, the split is determined by a subset A_j of categories that go to the left descendant node. Let N_{lL} and N_{lR} be the number of observations in the left descendant node and the right descendant node, and let $\hat{G}_{lL}$, $\hat{G}_{lR}$ be the corresponding Gini indices. Then we can define the *Gini gain*, an estimate of the improvement in impurity from this split, as $\nabla\hat{G} = \hat{G}_l - \frac{N_{lL}}{N_l}\hat{G}_{lL} - \frac{N_{lR}}{N_l}\hat{G}_{lR}$.

We search among all possible splits for all selected predictor variables, and select the split that generates the largest Gini gain.

1.2 Cforest

Conditional inference forests (Cforest), developed by Torsten Hothorn et al. in 2006 [8], can be considered as an alternative version of the original random forests. It is also an ensemble method that combines a collection of trees. Unlike the original random forests, either a bootstrap sample or a random sub-sample as large as 2/3 of the learning data may

be used for building the tree. Further, instead of using a CART tree, Cforest employs Ctree, an implementation of conditional inference trees, as the base learner. When fitting a Ctree, we derive the conditional distribution of the statistics that measure the associations between the response variable and the predictor variables. Multiple testing procedures (adjusted P-values) may be applied to determine whether there exist statistically significant associations between any of the predictors and the response. If yes, the predictor variable that is most strongly associated with the response will be selected for splitting; otherwise the recursive procedure will stop.

Following the notation in **Algorithm 1**, we can derive a series of hypothesis tests for testing $H_0^j : Y$ is independent of X_j, where $j \in \{1, 2, ..., p\}$. A global null hypothesis for testing the existence of association between any of the predictors $\{X_j\}_{j=1}^p$ and the response Y can be defined as $H_0 = \bigcap_{j=1}^m H_0^j$. Let α be the pre-specified significance level.

We define σ_t as one permutation of Y, and $\mathcal{P} = \{\sigma_t\}$ as all the possible permutations of Y.

Assume that Y has K different levels, let $B_k = \{i|y_i = k, i = 1, ..., N\}$, $k = 1, ..., K$. Similarly, for categorical predictor X_j, assume that X_j has L_j different levels, let $B_{jl_j} = \{i|x_{ij} = l_j, i = 1, ..., N\}$, $l_j = 1, ..., L_j$.

We can further derive a series of vectors R_j that measure the association between Y and X_j in the following way.

If X_j is continuous,

$$R_j = (\sum_{i \in B_1} x_{ij}, \sum_{i \in B_2} x_{ij}, ..., \sum_{i \in B_K} x_{ij})^T.$$

If X_j is discrete (categorical),

$$R_j = (\underbrace{n_{11}...n_{1L_j}}_{L_j}, \underbrace{n_{21}...n_{2L_j}}_{L_j}, ..., \underbrace{n_{K1}...n_{KL_j}}_{L_j})^T,$$

where n_{kl_j} is the size of $B_k \cap B_{jl_j}$.

Let $\mu_j = E(R_j|\mathcal{P})$ and $\Sigma_j = v(R_j|\mathcal{P})$ be the conditional expectation and conditional covariance of R_j, conditional on all possible permutations of Y.

Define the univariate test statistic U_j as $U_j = U(R_j, \mu_j, \Sigma_j) = (R_j - \mu_j)^T \Sigma_j^{-1} (R_j - \mu_j)$, where Σ_j^{-1} is the inverse or Moore-Penrose general inverse of Σ_j.

Here is an illustration of Cforest for classification.

Algorithm 2 : Cforest for Classification

---

1. For $b = 1$ to B:

   (a) Draw a sample $\mathcal{S}_b$ (a bootstrap sample or a random sub-sample containing 2/3 of the learning data) from the learning data $\mathcal{L}$.

   (b) Grow a binary Ctree T_b' on $\mathcal{S}_b$, by recursively repeating the following steps.

      i. Randomly select m variables from the p predictor variables.

      ii. Derive the corresponding vectors R_j for the selected variables.

      iii. Conduct all possible permutations of Y, compute μ_j, Σ_j and $U(R_j, \mu_j, \Sigma_j)$.

      iv. Based on $U(R_j, \mu_j, \Sigma_j)$, compute the P-values for testing H_0^j, $p_j = P(H_0^j)$.

      v. Compare $p = \min\{p_j\}$ to the pre-specified level α.

         - If $p \geq \alpha$, stop splitting.
         - If $p < \alpha$, select the predictor X_{j^*} that has the minimum p_j for splitting.

2. Output the ensemble of trees $\{T_b'\}_1^B$.

To make a prediction at a new point $\mathbf{x}$:

Let $\hat{C}_b'(\mathbf{x})$ be the prediction of the bth tree at $\mathbf{x}$, then $\hat{C}_{cf}^B(\mathbf{x}) =$ majority vote $\{\hat{C}_b'(\mathbf{x})\}_1^B$.

1.3 Permutation Variable Importance

Variable importance can be considered as a measure of association between the predictor variables and the response variable. It is used as a means of variable selection in many applications.

An advanced measure for variable importance that is adopted by both the original random forests and Cforest is called permutation variable importance, which is calculated from the out-of-bag data [9]. Since each tree is grown from a bootstrap sample (or a random sample containing about 2/3 of the learning data), on average, around 1/3 of the observations in the learning data are left out of the bootstrap sample (or subsample) and are not used in constructing the tree. These observations are considered as the out-of-bag (OOB) data for this tree. Following the previous notation, we further define $\mathcal{O}^b$ as the OOB data for the bth tree, where $b \in \{1, ..., B\}$ and the size of this OOB data is $|\mathcal{O}^b|$; let $C(\mathbf{x}_n) = y_n$ be the response of observation $\mathbf{z}_n$; let $\hat{C}_b(\mathbf{x}_n)$ be the predicted response value of observation $\mathbf{z}_n$ from the bth tree before permutation, and $\hat{C}_b^j(\mathbf{x}_n)$ be the predicted value from the bth tree after randomly permuting variable X_j over all OOB observations of that tree.

The permutation importance in Cforest is computed as follows. When the bth tree is grown, the OOB observations are passed down, and the OOB prediction accuracy (i.e. the percentage of observations classified correctly) is recorded. Then we randomly permute the values for variable X_j over all OOB observations of that tree. The permuted variable X_j', together with the remaining non-permuted predictor variables, is used to predict the response for the OOB observations, and the OOB prediction accuracy is again recorded. We use the difference in OOB prediction accuracy before and after permuting X_j, averaged over all trees, as a measure for variable importance [10] [11].

The computing of this permutation variable importance can be formalized as follows.

The variable importance of predictor variable X_j in the bth tree is:

$$VI^b(X_j) = \frac{1}{|\mathcal{O}^b|} \sum_{\mathbf{z}_n \in \mathcal{O}^b} I(\hat{C}_b(\mathbf{x}_n) = C(\mathbf{x}_n)) - \frac{1}{|\mathcal{O}^b|} \sum_{\mathbf{z}_n \in \mathcal{O}^b} I(\hat{C}_b^j(\mathbf{x}_n) = C(\mathbf{x}_n)).$$

Averaging over all trees, the permutation variable importance for predictor variable X_j in Cforest is:

$$VI(X_j) = \frac{1}{B} \sum_{b=1}^{B} VI^b(X_j).$$

We notice that although each $\mathcal{O}^b$ contains roughly about 1/3 of the total training data, the exact size of OOB data varies. While averaging the variable importance over all trees, the weights of each observation are actually slightly different due to the variation in $\mathcal{O}^b$. To fix this problem, the permutation variable importance in random forests is computed in a different manner.

In random forests, each observation $\mathbf{z}_n$ is in the OOB data in around 1/3 of all the trees. Assume that $\mathbf{z}_n$ is OOB in B_n trees, and let $\mathcal{T}_n$ be the indices of such trees, defined as $\mathcal{T}_n = \{b \in \{1, 2, ..., B\} | \text{observation } n \text{ is in the OOB data in tree } b\}$, it is obvious that $B_n = |\mathcal{T}_n|$. Within each tree that has $\mathbf{z}_n$ OOB, we randomly permute the values of X_j over all OOB observations of that tree. The differences of predictions before and after permutation are recorded. We average the prediction differences over all the trees such that $\mathbf{z}_n$ is OOB, and this is the variable importance of predictor X_j for observation $\mathbf{z}_n$. By averaging this variable importance over all the observations, we have obtained the permutation variable importance for X_j in random forests [12].

We can formalize the permutation variable importance in random forests as follows.

The variable importance of predictor variable X_j in observation $\mathbf{z}_n$ is:

$$VI_n(X_j) = \frac{1}{|\mathcal{T}_n|} \sum_{b \in \mathcal{T}_n} \{I(\hat{C}_b(\mathbf{x}_n) = C(\mathbf{x}_n)) - I(\hat{C}_b^j(\mathbf{x}_n) = C(\mathbf{x}_n))\}.$$

The permutation variable importance of predictor X_j in random forests is:

$$VI(X_j) = \frac{1}{N} \sum_{n=1}^{N} VI_n(X_j).$$

1.4 Comparison of Random Forests and Cforest

The main difference between random forests and Cforest is the different base learner employed by the two methods. Random forests are based on CART trees, while Cforests are built from conditional inference trees.

The CART tree has been criticized for its bias in variable selection (Dobra 2001 [13], Kim 2001 [14], Loh 1997 [15]). In the CART tree, the Gini gains for continuous predictors

are computed for all possible cutpoints within the range of the predictor variable. For categorical predictors, the Gini gains are computed for all possible ways of sending some categories to the left and the remaining categories to the right. The variable selected for the next split is the one that produces the highest Gini gain value. Variables with continuous scales or with more categories, and hence having more potential splits, are more likely to produce higher Gini gain values by chance, compared to categorical variables with fewer categories. Therefore, continuous variables and variables with more categories are artificially preferred for splitting in a CART tree, when none of the predictor variables are associated with the response variable, or when the associations are quite minor, or when the associations are equally strong.

The conditional inference trees, that are used for constructing Cfroest, are supposed to be unbiased because here the variable selection is conducted by minimizing the P-value of a conditional inference independence test, which is compared to a χ^2 test that incorporates the number of categories of each variable in the degree of freedom [16].

Torsten Hothorn and Carolin Strobl et al., the developers of Cforest, claimed in Strobl 2007 [17] that the bias of variable selection for splitting in each individual CART tree has directly weakened the classification capacity of random forests. Further, they mentioned that the hypothesis testing adopted by Cforest is not reliable when bootstrap samples are used, which will essentially affect the credibility of the permutation variable importance in Cforest. Hothorn and Strobl have also suggested that the variable importance measure available in Cforest, together with sub-sampling without replacement, should be used to achieve accurate classifications and reliable estimates of the variable importance.

However, our finding is that the bias of variable selection for splitting in each individual CART tree does not have a severe impact on the performance of random forests. We conjecture that this is due to a number of possible reasons. The preference for continuous variables and variables with more categories is significant only under the situations when the predictor variables are not associated with the response variable, or the association is too weak to produce any significant improvement in impurity from splitting that predictor

variable. Under these situations, the applicability of random forests or any other methods is doubtful since the information contained in the data is quite limited. If there exist any detectable associations that can be used for reasonable splits based on either Gini criterion or independence tests, the effect of association will exceed the effect of biased selection preference. Furthermore, by choosing from only a small subset of all predictor variables at each split, the bias of variable selection in each individual tree has been reduced. The instabilities of individual CART trees will also be offset by the ensemble procedure. Therefore, the performance of random forests will still be superior, even compared to Cforest. To prove these conjectures is beyond the scope of this project.

In the next section, we will systematically compare random forests with Cforest in several simulation studies.

Chapter 2

Simulation Studies and Results

The results in this paper were obtained by using add-on packages "randomForest 4.5-33" [18] and "party 0.9-999" [19] in R 2.9.2 [20].

The simulation designs used throughout this paper represent scenarios where a binary response variable Y is predicted from a set of independent potential predictor variables that vary in the scale of measurement and level of categories.

The construction of both random forests and Cforest is greatly affected by the values of tuning parameters. The parameter "ntree" determines the size of the forests, that is the number of trees to grow. The default value of "ntree" is 500. In our studies, this was set to be either 500 or 1000. The parameter "mtry" controls the number of variables randomly selected as candidates at each split. When there are p predictor variables, the default value of "mtry" is $\sqrt{p}$. We tried all feasible values of "mtry", which ranged from 1 to p. The parameter "replace" decides the scheme of subsampling from the learning data: when replace=TRUE, it is sampling with replacement and this is the default bootstrap sample adopted by random forests; when replace=FALSE, it is sampling without replacement recommended by the creators of Cforest. For sampling without replacement the subsample size is set to 0.632 times the original learning data size, because in bootstrap sampling without replacement about 63.2% of the original data end up in the bootstrap sample.

Under every combination of the three tuning parameter values, both random forests and Cforest were built from a set of learning data which had 100 observations. Then we applied the fitted models to predict the response variable on a set of testing data that had $1,000$ observations, where the testing data were generated from the same distribution as the learning data. The misclassification rates on the testing data were recorded as the measure of predictive performance. These steps were repeated over 100 simulation runs,

the mean misclassification rates were considered as the representations of the predictive power. The tuning parameter values that led to the minimum mean misclassification rate were chosen as the optimal values. In order to examine the effect of bootstrap sampling, "replace" was set to be either TRUE or FALSE. The optimal "mtry" and "ntree", together with the prespecified "replace", are used to compute the permutation importance measures for all predictor variables, which were illustrated via boxplots and parallel coordinate plots. Furthermore, over the 100 simulation runs, the trials in which informative predictor variables were correctly distinguished were recorded. The unsuccessful trials were highlighted with blue in the parallel plots.

2.1 Simulation Study I

2.1.1 Simulation Design

The first simulation study deals with the "XOR" problem. In this example, all predictor variables are sampled independently from the distributions in Table 2.1. $U(-1,1)$ stands for the continuous uniform distribution with range from -1 to 1. $N(0,1)$ stands for the standard normal distribution. $M(k)$ stands for the multinomial distribution with values $\{1,2,...,k\}$ and equal probabilities (discrete uniform distribution on $\{1,2,...,k\}$).

Table 2.1: Simulation Design I - Predictor Variables

Predictor Variables		
X_1	$\sim$	$U(-1,1)$
X_2	$\sim$	$U(-1,1)$
X_3	$\sim$	$M(2)$
X_4	$\sim$	$M(20)$
X_5	$\sim$	$U(-1,1)$
X_6	$\sim$	$N(0,1)$

The response variable Y depends on the first two predictor variables, as defined in Table 2.2.

Table 2.2: Simulation Design I - Response Variable

Response Variable	
$Y=0$	if $X_1X_2 \leq 0$
$Y=1$	if $X_1X_2 > 0$

2.1.2 Results: Prediction Accuracies

In this example, there are 24 different combinations of tuning parameter values. Under each combination, the mean misclassifation rate and the standard error for both functions, over 100 simulation runs, are listed in Table 2.3. The mean misclassification rates differ dramatically as "mtry" changes. The choice of "replace" will also significantly affect the error rates, usually replace=TRUE leads to better predictions. Random forests are consistently better than Cforest in prediction accuracies. In both functions, the minimum mean misclassfication error rate is achieved when mtry=6, ntree=1000 and replace=TRUE.

Table 2.3: Simulation Design I - Prediction Accuracies for 100 Runs

mtry	ntree	replace	RF error rate	CF error rate	RF std error	CF std error
1	500	FALSE	0.3431	0.4289	0.0034	0.0033
1	500	TRUE	0.3233	0.3891	0.0038	0.0041
1	1000	FALSE	0.3411	0.4270	0.0037	0.0036
1	1000	TRUE	0.3237	0.3907	0.0034	0.0039
2	500	FALSE	0.2054	0.3765	0.0053	0.0051
2	500	TRUE	0.1885	0.3080	0.0047	0.0054
2	1000	FALSE	0.2054	0.3750	0.0052	0.0052
2	1000	TRUE	0.1865	0.3047	0.0045	0.0053
3	500	FALSE	0.1561	0.3431	0.0059	0.0066
3	500	TRUE	0.1404	0.2586	0.0052	0.0070
3	1000	FALSE	0.1546	0.3422	0.0061	0.0067
3	1000	TRUE	0.1401	0.2615	0.0051	0.0067
4	500	FALSE	0.1265	0.3119	0.0063	0.0085
4	500	TRUE	0.1134	0.2215	0.0053	0.0079
4	1000	FALSE	0.1261	0.3143	0.0064	0.0085
4	1000	TRUE	0.1129	0.2230	0.0054	0.0079
5	500	FALSE	0.1076	0.2880	0.0064	0.0098
5	500	TRUE	0.0952	0.1891	0.0054	0.0085
5	1000	FALSE	0.1070	0.2867	0.0064	0.0099
5	1000	TRUE	0.0950	0.1924	0.0054	0.0088
6	500	FALSE	0.0955	0.2647	0.0065	0.0112
6	500	TRUE	0.0866	0.1693	0.0055	0.0090
6	1000	FALSE	0.0961	0.2638	0.0065	0.0111
6	**1000**	**TRUE**	**0.0847**	**0.1661**	**0.0053**	**0.0091**

2.1.3 Results: Permutation Variable Importance Measures

The optimal tuning parameter values with which the best prediction accuracy is obtained should be used for computing permutation variable importance. Thus mtry=6, ntree=1000 and replace=TRUE, as well as mtry=6, ntree=1000 and replace=FALSE are used in this example.

Figure 2.1 shows boxplots of the distributions of the permutation variable importance measures over 100 simulation runs. The distributions are not significantly different with the choice of "replace". In each plot, the mean permutation importances of the informative variables X_1 and X_2 are roughly the same, significantly higher than those of the uninformative variables X_3, X_4, X_5 and X_6, which are all approximately 0.

Despite the fact that the uninformative predictor variables vary in the scale of measurement and levels of categories, the variances of the permutation importances of the uninformative predictors are approximately the same in random forests. This indicates that the variable selection bias induced by the Gini Criterion does not severely affect the variance of the permutation importance measures. Further, the variances of the informative variables are smaller than those of the uninformative variables in random forests. The situation in Cforest is just the opposite, informative predictors have higher variances.

To further explore the permutation variable importances, we look at the parallel coordinate plots in Figure 2.2. In parallel plots for random forests, the permutation importance measures of informative variables are consistently higher than uninformative variables over 100 simulation runs. For Cforest, when replace=FALSE as recommended in Strobl 2007 [17], in 8 out of 100 trials, the permutation importance of one or both of the informative variables is misleadingly lower than that of at least one of the uninformative predictors.

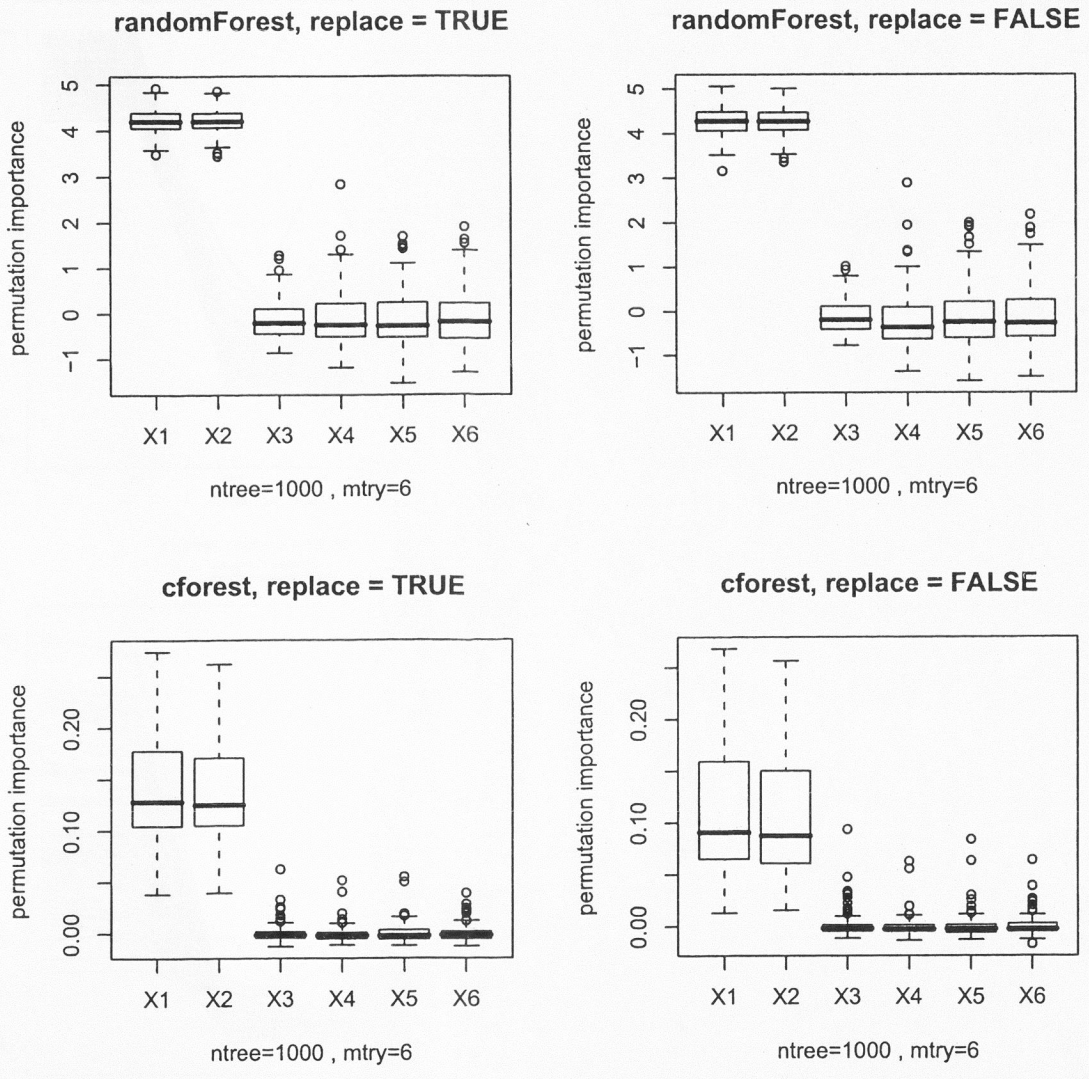

Fig. 2.1: Boxplots of the distributions of permutation variable importance measures from random forests and Cforest over 100 repeats in simulation study I.

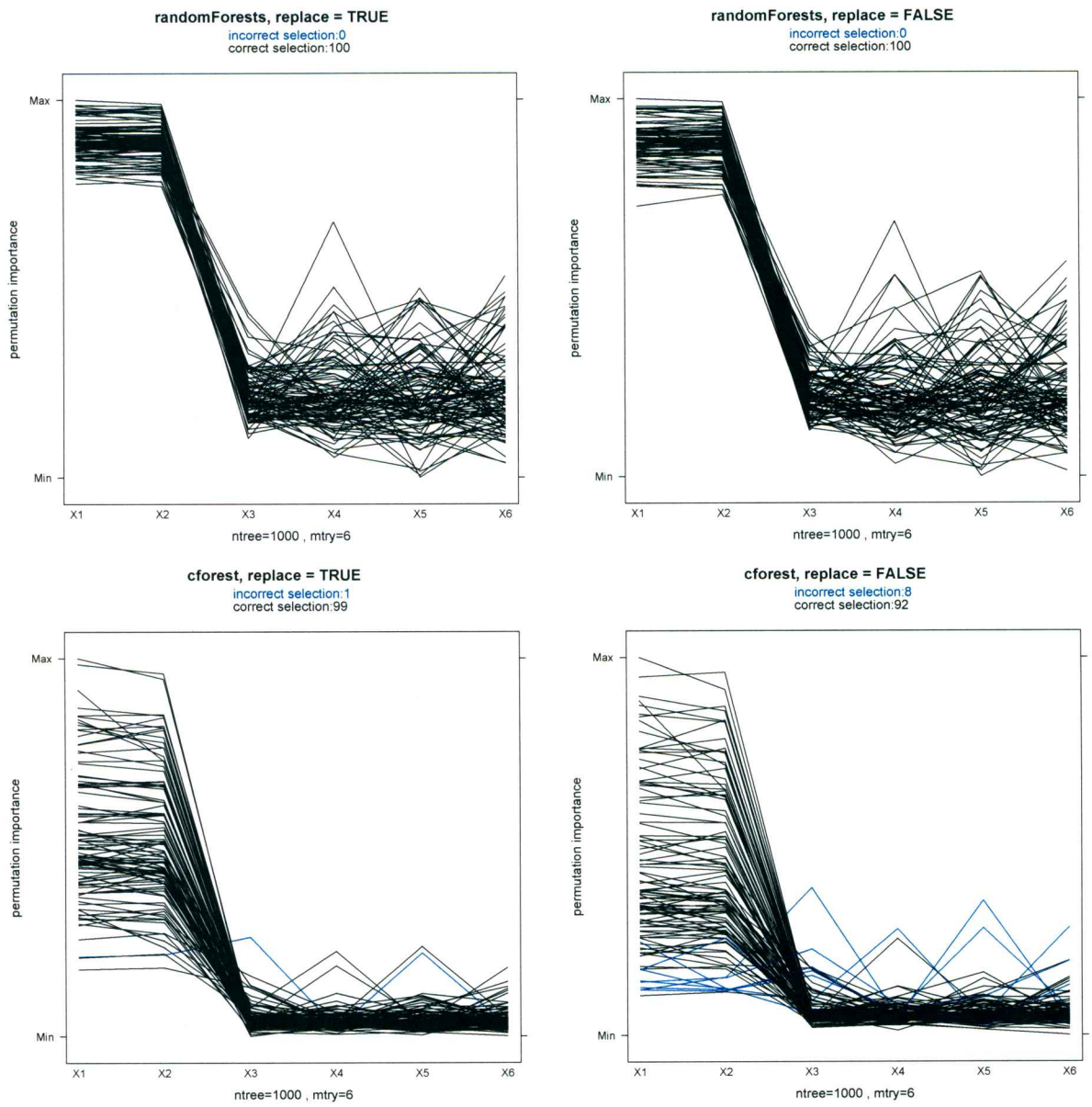

Fig. 2.2: Parallel coordinate plots of permutation variable importance measures from random forests and Cforest over 100 repeats in simulation study I; blue stands for incorrect selection.

2.2 Simulation Study II

2.2.1 Simulation Design

In the second example, 11 predictor variables are sampled independently from the distributions in Table 2.4.

Table 2.4: Simulation Design II - Predictor Variables

Predictor Variables		
X_1	$\sim$	$M(2)$
X_2	$\sim$	$M(2)$
X_3	$\sim$	$M(2)$
X_4	$\sim$	$M(2)$
X_5	$\sim$	$M(2)$
X_6	$\sim$	$M(4)$
X_7	$\sim$	$M(10)$
X_8	$\sim$	$M(20)$
X_9	$\sim$	$U(-1,1)$
X_{10}	$\sim$	$U(-4,4)$
X_{11}	$\sim$	$N(0,1)$

The response variable Y depends on the first four predictor variables, as defined in Table 2.5.

Table 2.5: Simulation Design II - Response Variable

Response Variable	
$Y=0$	if $X_1+X_2+X_3+X_4 \leq 6$
$Y=1$	if $X_1+X_2+X_3+X_4 > 6$

2.2.2 Results: Prediction Accuracies

There are 44 different combinations of tuning parameter values. The mean misclassifation rates and the standard errors are listed in Table 2.6. We can see that the predictive power of random forests is not affected by the dramatic differences in scale of measurement and level of categories among the predictor variables. Random forests have better prediction accuracies than Cforest. Random forests achieve their minimum error rate when mtry=5, ntree=500 and replace=FALSE. Cforest achieves its minimum error rate when mtry=6, ntree=1000 and replace=TRUE.

Table 2.6: Simulation Design II - Prediction Accuracies for 100 Runs

mtry	ntree	replace	RF error rate	CF error rate	RF std error	CF std error
1	500	FALSE	0.1944	0.2800	0.0059	0.0039
1	500	TRUE	0.1945	0.2605	0.0058	0.0050
1	1000	FALSE	0.1944	0.2812	0.0059	0.0037
1	1000	TRUE	0.1938	0.2615	0.0058	0.0050
2	500	FALSE	0.1122	0.2090	0.0043	0.0062
2	500	TRUE	0.1159	0.1844	0.0045	0.0061
2	1000	FALSE	0.1106	0.2113	0.0043	0.0060
2	1000	TRUE	0.1169	0.1846	0.0044	0.0061
3	500	FALSE	0.0923	0.1656	0.0038	0.0064
3	500	TRUE	0.0977	0.1431	0.0040	0.0057
3	1000	FALSE	0.0902	0.1665	0.0040	0.0064
3	1000	TRUE	0.0978	0.1452	0.0040	0.0057
4	500	FALSE	0.0843	0.1416	0.0037	0.0057
4	500	TRUE	0.0916	0.1226	0.0036	0.0049
4	1000	FALSE	0.0825	0.1400	0.0036	0.0058
4	1000	TRUE	0.0906	0.1221	0.0037	0.0049
5	**500**	**FALSE**	**0.0807**	0.1317	**0.0035**	0.0050
5	500	TRUE	0.0889	0.1140	0.0036	0.0046
5	1000	FALSE	0.0809	0.1292	0.0035	0.0050
5	1000	TRUE	0.0881	0.1133	0.0035	0.0046
6	500	FALSE	0.0813	0.1291	0.0034	0.0047
6	500	TRUE	0.0870	0.1122	0.0034	0.0046
6	1000	FALSE	0.0809	0.1272	0.0035	0.0048
6	**1000**	**TRUE**	0.0882	**0.1110**	0.0035	**0.0046**
7	500	FALSE	0.0829	0.1323	0.0035	0.0047
7	500	TRUE	0.0889	0.1132	0.0036	0.0045
7	1000	FALSE	0.0826	0.1350	0.0035	0.0047
7	1000	TRUE	0.0893	0.1136	0.0033	0.0046
8	500	FALSE	0.0854	0.1434	0.0034	0.0049
8	500	TRUE	0.0909	0.1138	0.0035	0.0046
8	1000	FALSE	0.0847	0.1408	0.0035	0.0048
8	1000	TRUE	0.0910	0.1160	0.0035	0.0047
9	500	FALSE	0.0867	0.1467	0.0035	0.0045
9	500	TRUE	0.0930	0.1194	0.0035	0.0047
9	1000	FALSE	0.0872	0.1477	0.0035	0.0046
9	1000	TRUE	0.0918	0.1189	0.0034	0.0048
10	500	FALSE	0.0896	0.1514	0.0037	0.0043
10	500	TRUE	0.0952	0.1216	0.0036	0.0047
10	1000	FALSE	0.0894	0.1515	0.0037	0.0044
10	1000	TRUE	0.0951	0.1222	0.0036	0.0048
11	500	FALSE	0.0942	0.1539	0.0039	0.0042
11	500	TRUE	0.0981	0.1245	0.0037	0.0045
11	1000	FALSE	0.0928	0.1517	0.0038	0.0044
11	1000	TRUE	0.0973	0.1263	0.0036	0.0047

2.2.3 Results: Permutation Variable Importance Measures

In random forests, the optimal tuning parameter values used for computing permutation variable importance are mtry=5 and ntree=500, while mtry=6 and ntree=1000 are used by Cforest.

Figure 2.3 shows boxplots of the distributions of the permutation variable importance measures over 100 simulation runs. The distributions are not significantly different with the choice of parameter "replace". As we expected, the mean permutation importances of the informative variables X_1, X_2, X_3 and X_4 are roughly the same, significantly higher than those of the uninformative variables, which are all approximately 0.

The variances of the permutation importance are approximately the same in random forests. Thus we think the variances of the permutation variable importance measures are not affected by the differences in scale of measurement and level of categories among the predictor variables. In Cforest, the variances of uninformative variables are extremely small. However, the variances of informative variables are so large that I would anticipate some errors in ranking in single trials.

The parallel coordinate plots in Figure 2.4 confirmed my intuition. While random forests were able to successfully distinguish the informative predictors in all 100 trials, Cforest failed once with both replace=TRUE and replace=FALSE.

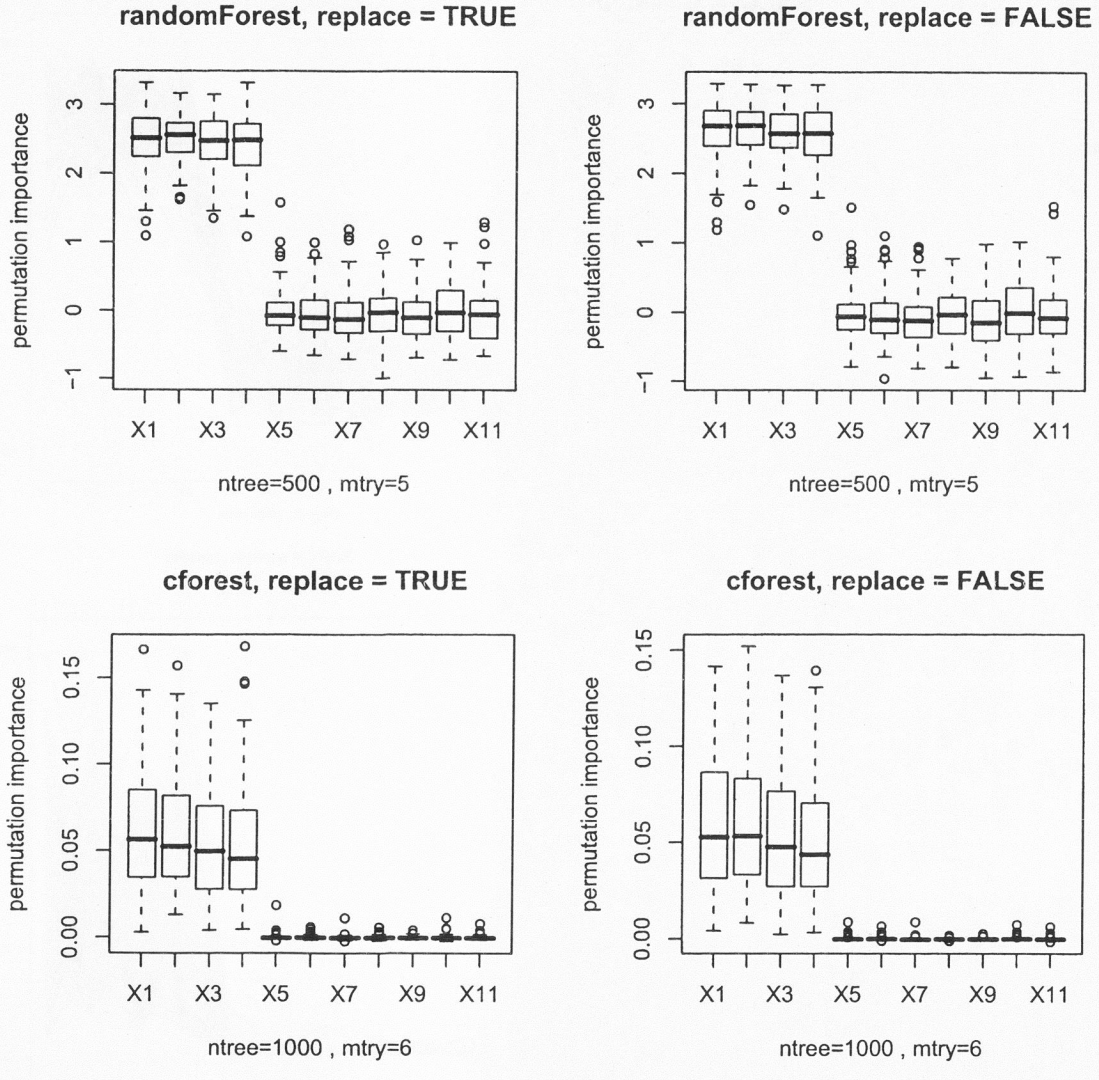

Fig. 2.3: Boxplots of the distributions of permutation variable importance measures from random forests and Cforest over 100 repeats in simulation study II.

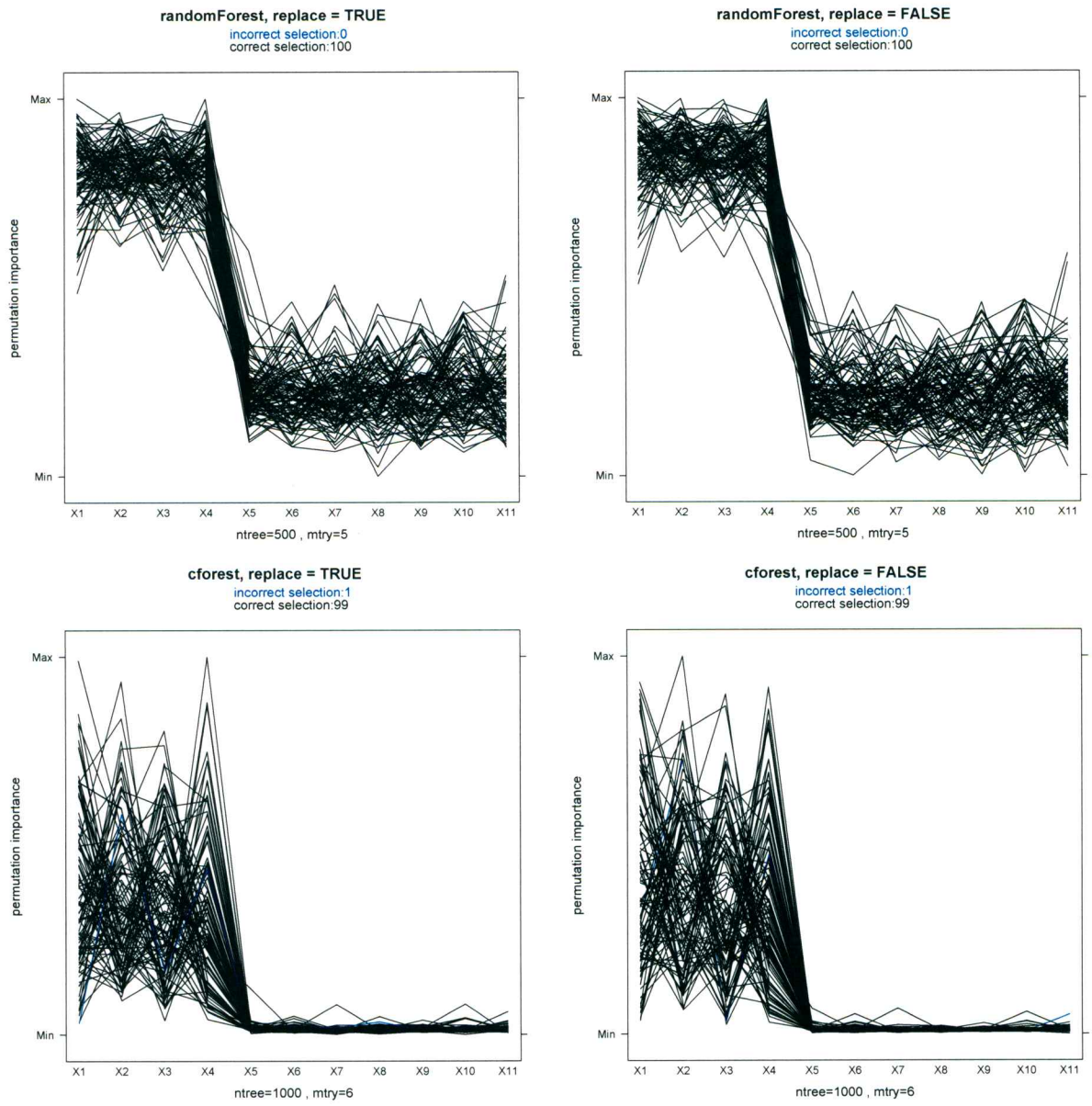

Fig. 2.4: Parallel coordinate plots of permutation variable importance measures from random forests and Cforest over 100 repeats in simulation study II; blue stands for incorrect selection.

2.3 Simulation Study III

2.3.1 Simulation Design

The third example was used in Strobl 2007 [17] as an illustration that random forests was not as good as Cforest. Here the predictor variables are sampled independently from the distributions in Table 2.7.

Table 2.7: Simulation Design III - Predictor Variables

Predictor Variables		
X_1	$\sim$	$M(2)$
X_2	$\sim$	$M(4)$
X_3	$\sim$	$M(10)$
X_4	$\sim$	$M(20)$
X_5	$\sim$	$U(-1,1)$
X_6	$\sim$	$N(0,1)$

The response variable Y only depends on predictor X_1. The degree of dependence between Y and X_1 is regulated by the distribution of Y conditional on X_1, as defined in Table 2.8. $B(p)$ stands for the binomial distribution. We realize that the association between Y and X_1 is relatively weak here, compared to the previous examples.

Table 2.8: Simulation Design III - Response Variable

Response Variable		
$Y\|X_1 = 1$	$\sim$	$B(0.3)$
$Y\|X_1 = 2$	$\sim$	$B(0.7)$

2.3.2 Results: Prediction Accuracies

Twenty four different combinations of tuning parameter values exist in this example. Under each combination, the mean misclassifation rate and the standard error for both functions are listed in Table 2.9. The prediction accuracies of Cforest are uniformly better than those of random forests over all tuning parameter combinations. The lowest misclassification error rate for random forests is 0.3474, and it is obtained when mtry=1, ntree=1000 and replace=TRUE. The lowest error rate for Cforest is 0.3190, and it is obtained when

mtry=4, ntree=1000, replace=FALSE. The lowest error rate for Cforest is only slightly better than that for random forests. Rarely, the prediction accuracies decrease with increasing "mtry" for random forests in this example. My explanation for this phenomenon is that the relevance between Y and X_1 is too weak. When more than one predictor variables are randomly chosen at each split, this weak association will be overwhelmed by the selection bias induced by the Gini Criterion. However, the predictive power of Cforest is also dubious due to the roughly 30% misclassification rate.

Table 2.9: Simulation Design III - Prediction Accuracies for 100 Runs

mtry	ntree	replace	RF error rate	CF error rate	RF std error	CF std error
1	500	FALSE	0.3479	0.3538	0.0034	0.0049
1	500	TRUE	0.3481	0.3392	0.0035	0.0041
1	1000	FALSE	0.3474	0.3516	0.0033	0.0049
1	**1000**	**TRUE**	**0.3474**	0.3383	**0.0034**	0.0042
2	500	FALSE	0.3760	0.3317	0.0037	0.0039
2	500	TRUE	0.3733	0.3353	0.0038	0.0038
2	1000	FALSE	0.3755	0.3320	0.0037	0.0040
2	1000	TRUE	0.3727	0.3348	0.0038	0.0037
3	500	FALSE	0.3767	0.3225	0.0037	0.0036
3	500	TRUE	0.3743	0.3292	0.0038	0.0036
3	1000	FALSE	0.3766	0.3222	0.0038	0.0036
3	1000	TRUE	0.3742	0.3295	0.0038	0.0035
4	500	FALSE	0.3782	0.3196	0.0038	0.0035
4	500	TRUE	0.3778	0.3281	0.0039	0.0035
4	**1000**	**FALSE**	0.3783	**0.3190**	0.0038	**0.0034**
4	1000	TRUE	0.3771	0.3282	0.0039	0.0036
5	500	FALSE	0.3812	0.3192	0.0038	0.0034
5	500	TRUE	0.3800	0.3296	0.0038	0.0036
5	1000	FALSE	0.3806	0.3191	0.0038	0.0034
5	1000	TRUE	0.3799	0.3301	0.0038	0.0035
6	500	FALSE	0.3835	0.3206	0.0037	0.0035
6	500	TRUE	0.3815	0.3310	0.0038	0.0035
6	1000	FALSE	0.3831	0.3200	0.0038	0.0034
6	1000	TRUE	0.3819	0.3315	0.0038	0.0036

2.3.3 Results: Permutation Variable Importance Measures

The optimal tuning parameter values mtry=1, ntree=1000 and replace=TRUE are used for computing the permutation variable importances in random forests. To examine the effect of bootstrap sampling, we have also included the permutation importance computed when replace=FALSE. For Cforest, the permutation importance is computed with mtry=4, ntree=1000.

Figure 2.5 shows boxplots of the distributions of the permutation variable importance measures over 100 simulation runs. Despite the fact that the relevance of Y and X_1 is weak and the prediction accuracies are not satisfying, the permutation variable importance measures still maintain reasonably good performances, given the optimal tuning parameter values. The mean permutation importances of the uninformative variables are approximately 0, and the mean permutation importance of the informative variable X_1 is significantly larger than 0.

In random forests, the variances of the permutation importance are roughly the same for all predictor variables, which further convinces me that the permutation variable importance measures are not affected by the differences in scale of measurement and level of categories among predictor variables.

The parallel coordinate plots in Figure 2.6 show that in 100 simulation runs, both random forests and Cforest fail only twice in distinguishing the informative predictor variable, regardless of subsampling with or without replacement.

Comparing the plots with replace=TRUE to the plots with replace=FALSE, only minor differences that can be considered as randomness are spotted. Thus I am also further convinced that the bootstrap sampling scheme does not have harmful effects on permutation importance measures.

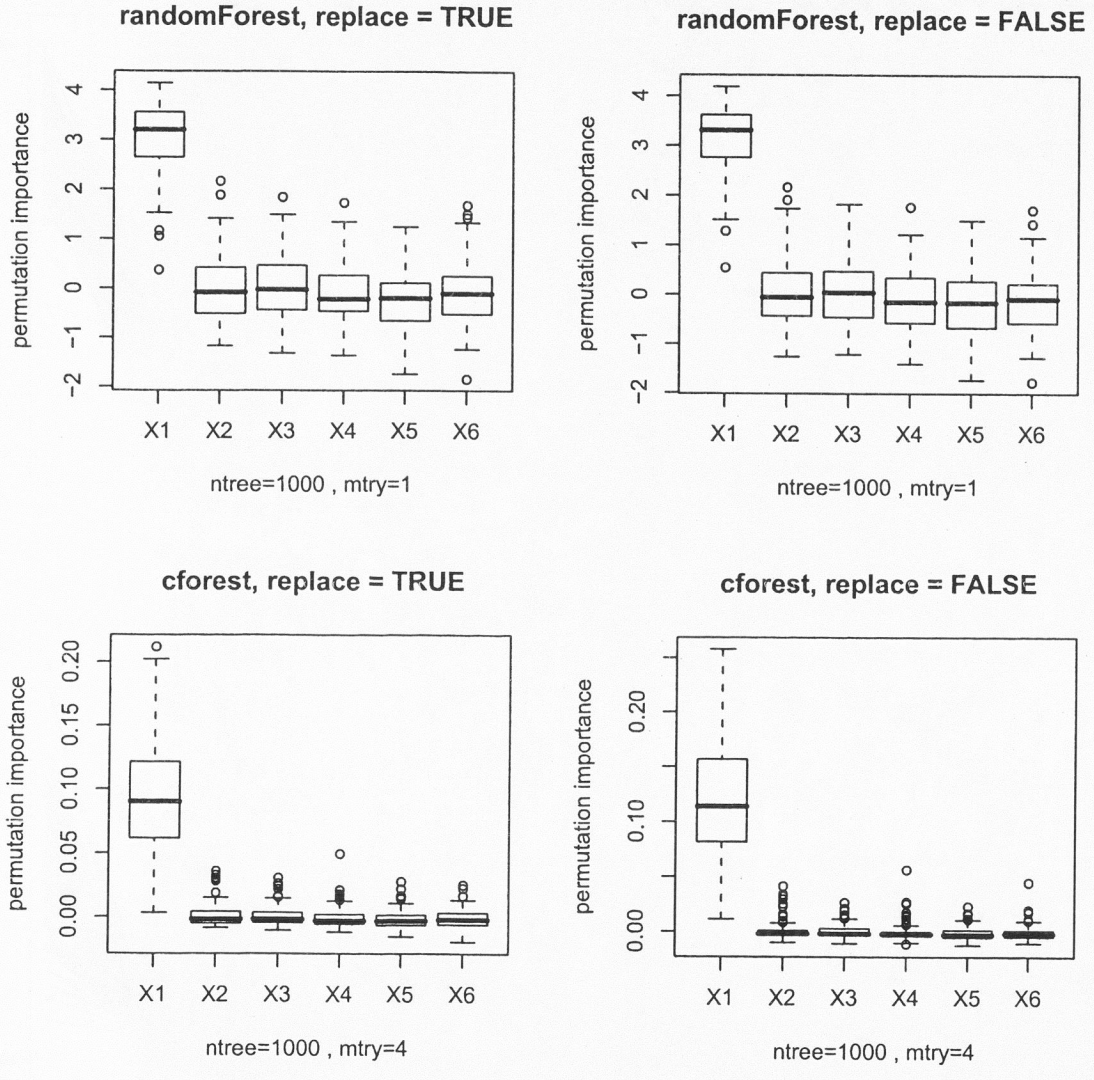

Fig. 2.5: Boxplots of the distributions of permutation variable importance measures from random forests and Cforest over 100 repeats in simulation study III.

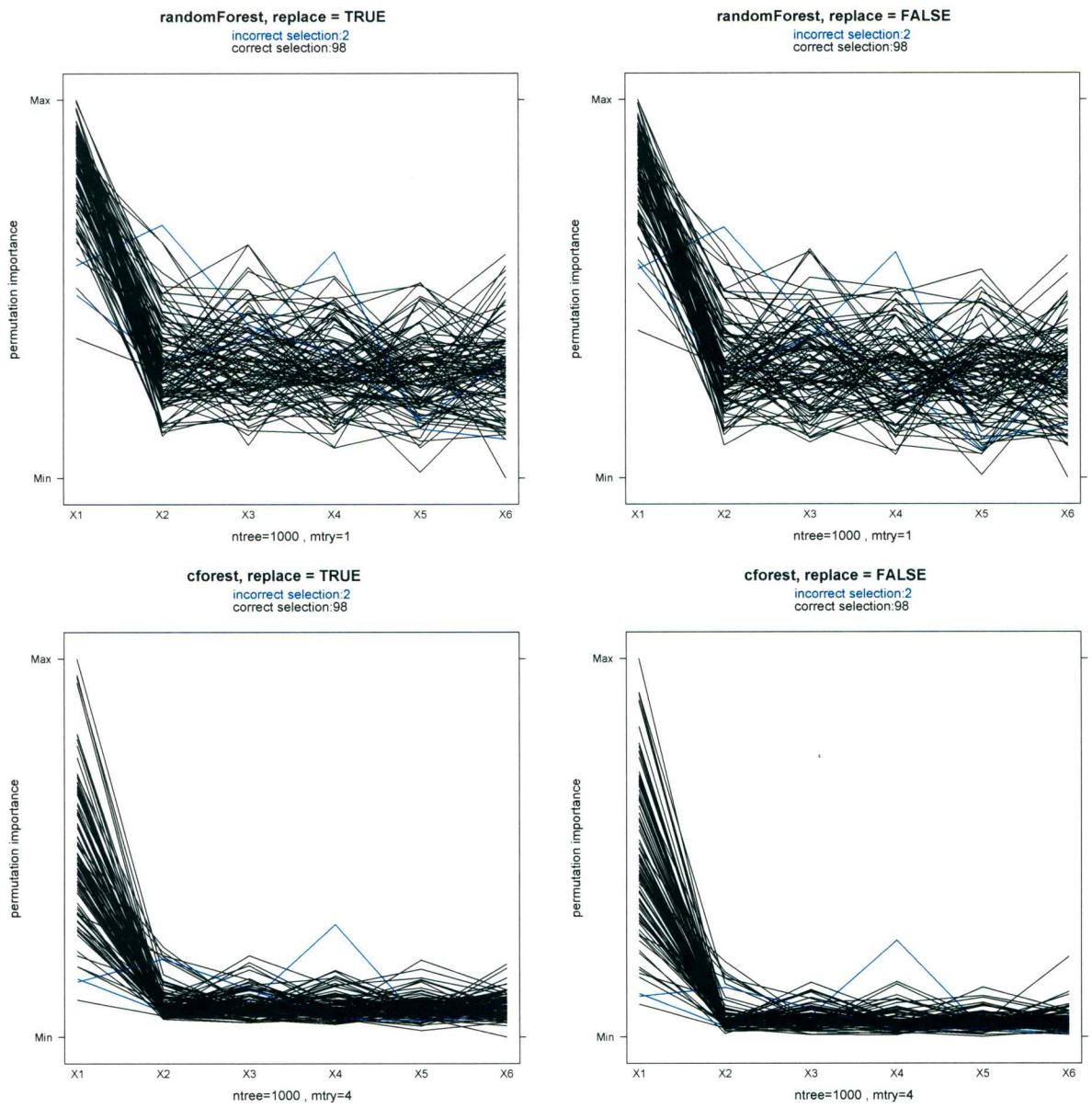

Fig. 2.6: Parallel coordinate plots of permutation variable importance measures from random forests and Cforest over 100 repeats in simulation study III; blue stands for incorrect selection.

Chapter 3

Conclusion

Random forests is a powerful statistical tool that provides accurate predictions and reliable variable importance measures. Recently, criticism has been leveled at random forests for the variable selection bias induced by the Gini Criterion. A new implementation of random forests, called Cforest, was developed based on conditional inference trees. The creators of Cforest have constructed extreme simulations that exaggerate the variable selection bias of random forests and claimed that random forests should be replaced by Cforest.

Through more systematic analysis and more general simulation studies, we have shown that the predictive power and permutation variable importance measure in random forests are not severely affected by the variable selection bias induced by the Gini Criterion.

In most situations where a modest association between the response variable and the predictor variables exists, with proper setting of tuning parameters, random forests is able to provide more accurate predictions and more reliable permutation variable importance measures compared to Cforest.

Only under extreme situations, when the relevance between the response and the predictors is weak, is Cforest able to achieve better results, and these results are usually unimpressive because their prediction accuracies are low.

So we should consider Cforest as a sensible complement, rather than a superior surrogate of random forests.

One more finding is that the default tuning parameter values usually are not the best options. Users should try different values to get better performance, especially for "mtry".

Chapter 4

Discussion and Future Work

In this paper we have shown empirical evidence that random forests do not suffer from the criticisms on the variable selection bias induced by the Gini Criterion. Further research will aim at providing a more rigorous explanation under the theoretical framework [21].

Further, although the predictive power and permutation variable importance measure for Cforest is not as good as random forests in general situations, its capacity in detecting weak associations deserves more exploration. Creating a better splitting criterion that could account for different scales of measurement and different numbers of categories which will improve the performance of random forests is the ultimate target.

References

[1] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees.* New York: Chapman and Hall, 1984.

[3] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Date Mining, Inference, and Prediction.* Springer, 2001.

[4] A. Cutler, "Random forests," *Encyclopedia of Statistics in Behavioral Science*, vol. 4, pp. 1665–1667, 2005.

[5] A. Cutler and J. R. Stevens, *Random Forests for Microarrays*, ser. Methods in Enzymology, vol. 411, ch. DNA Microarrays, Part B: Databases and Statistics. Academic Press, 2006, editor Alan R. Kimmel and Brian Oliver.

[6] A. Cutler, D. R. Cutler, and J. R. Stevens, *Tree Based Methods*, ch. High-Dimensional Data Analysis in Oncology. Springer, 2008, editor Xiaochun Li and Ronghui Xu.

[7] C. Strobl, A. L. Boulesteix, and T. Augustin, "Unbiased split selection for classification trees based on the gini index," *Computational Statistics and Data Analysis*, vol. 52, pp. 483–501, 2007.

[8] T. Hothorn, K. Hornik, and A. Zeileis, "Unbiased recursive partitioning: a conditional inference framework," *Journal of Computational and Graphical Statistics*, vol. 15, pp. 651–674, 2006.

[9] A. Cutler, "Bagging," *Encyclopedia of Statistics in Behavioral Science*, vol. 1, pp. 115–117, 2005.

[10] K. J. Archer and R. V. Kimes, "Empirical characterization of random forest variable importance measures," *Computational Statistics and Data Analysis*, vol. 52, 2007.

[11] C. Strobl, A. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC Bioinformatics*, vol. 9, 2008.

[12] Adele Cuter and Leo Breiman, "Random Forests Website," [http://www.math.usu.edu/ adele/forests/index.htm], 2009.

[13] A. Dobra and J. Gehrke, "Bias correction in classification tree construction," in *In Proceedings of the Seventeenth International Conference on Machine Learning, Williams College, Williamstown, MA, USA*, pp. 90–97, 2001.

[14] H. Kim and W. Loh, "Classification trees with unbiased multiway splits," *Journal of the American Statistical Association*, vol. 96, pp. 589–604, 2001.

[15] W. Loh and Y. Shih, "Split selection methods for classification trees," *Statistica Sinica*, vol. 7, no. 4, pp. 815–840, 1997.

[16] R. Miller and D. Siegmund, "Maximally selected chi square statistics," *Biometrics*, vol. 38, no. 4, pp. 1011–1016, 1982.

[17] C. Strobl, A. L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC Bioinformatics*, vol. 8, no. 25, 2007.

[18] L. Breiman, A. Cutler, A. Liaw, and M. Wiener, "Breiman and Cutler's Random Forests for Classification and Regression," [http://CRAN.R-project.org/], 2009.

[19] T. Hothorn, K. Hornik, and A. Zeileis, "party: A Laboratory for Recursive Part(y)itioning," [http://CRAN.R-project.org/], 2009.

[20] R Development Core Team, "R: A Language and Environment for Statistical Computing," [http://www.R-project.org/], 2009.

[21] M. van der Laan, "Statistical inference for variable importance," *International Journal of Biostatistics*, vol. 2, 2006.